Newsletter

June 1990

In this issue ...

- A3000 reviewed
- Introduction to Electronics
- A.M.A.S.
- The Shadow – a short story
- Rocco's Ramblings
- All the fun of the Fair

HONORARY CUGI OFFICIALS

Chairman	Geoffrey J. Reeves
Treasurer	Brian Ward
Membership Secretary	Shane Broadberry
Chief Librarian	Thomas Kinsella
Amiga Librarian	Rocco Matassa
PC Librarian	Brian Ward
C64 Librarian	Stephen Kemp
C128 Librarian	Geoffrey J. Reeves
Communications	Liam Murphy
Newsletter Editor	Edward Carroll

CUGI meeetings are held every second Friday, at 8:00 PM in the computer room at St. Andrew's College Booterstown. All correspondence should be addressed to:

CUGI
c/o St. Andrew's College,
Booterstown,
Blackrock,
Co. Dublin.

All articles in this newsletter are copyrighted by the author unless otherwise stated. If you wish to reprint an article, write to its author c/o CUGI at the above address.

Newsletter
Contents

Volume 2 Number 3
June 1990

Editorial	Geoffrey J. Reeves	2
Amiga, Mac or PC?	Brian Ward	4
Understanding Electronics, Part1	Stephen Kemp	6
Amiga 3000: First Impressions	Eddy Carroll	8
Fair Enough!	Geoffrey J. Reeves	12
A.M.A.S.	Brian Ward	16
Designer C	Shane Broadberry	18
The Shadow	Colm O'Rourke	25
Club News	Secretary	27
Rocco's Ramblings	Rocco Matassa	28
Stephanie's Problem Page	Stephanie	30

Editorial

I knew I shouldn't have joked about him joining the ranks of tax-payers. I should've known there'd be a catch – there always is. Eddy begins (real) work and has to be trained in the USA for five weeks. Gee, it's *so* tough out there. Shucks, I'm sure he'll have a great time – but what of the Newsletter? No problem, we'll have it done before he goes away ... well, nearly. Basically, I'm your friendly Editor again!

As it turns out, this issue is again smaller than the last. I hope that this is not a trend. I know that this little (?) magazine of ours *is* impressive – most groups such as ours cannot and do not produce anything like it. I know our members have been impressed ... very impressed so I am not going to begin to plead for submissions. After all, we have always said that this is *your* Newletter – you submit the articles, whatever the topic, whatever the size – right? Right!

Sadly, the hope of including a number of pictures has not yet become a reality. The fact that we would very much like a pictorial (front) cover is a key reason for pushing this idea. Mind you, it would help if we had a decent picture. What did you think of the stuff on the disk which came with *Amiga Computing* (August 1990)?

By the way, if you think that the utilities which come with these magazine disks are brilliant, why don't you ask our librarian for some disks? Recent disks have included NewZap and PopUpMenu, for example – we've had them for some time now. If you have an Amiga and you don't have the Fred Fish Catalogue (it's all on one disk – we'll give you a copy for nothing!), well you *are* a right wally. Actually, the catalogue is itself a Fish Disk (number 301) and it contains a program (*Aquarium*), a database and some utilities. Obviously, the program's author has a worse sense of humour than me ... keeping the fish in an aquarium!

The database can be kept up-to-date fairly easily and it allows you to hunt for programs, demos, utilities or whatever in any combination and it gives you a brief synopsis of each one it finds. There are many excellent utilities available – do you realise that there are still users who don't have Dmouse?!?! It really is sad ⌢

Anyway, enough of my rambling – what's in this issue? Well you've read the cover so you have an idea already. I was at *the 16-bit Computer Fair* in London and there's a report on that and the latest news gleaned from it. Brian reviews one of my purchases (A.M.A.S.) and Eddy reviews the A3000.

For the TEXnically (oops ... OZONE [1]), or rather, technically minded Steve has begun a series on electronics. This should be of interest to many who may know a little about software but not so much about hardware. Why not start at the beginning?

Our resident Agony Aunt continues to receive letters from people who are shy about airing their problems in public and a couple of you have your answer printed herein. I hope it helps. C programmers (I'm one) know all about the effect of bad programming, poor style, etc. and Shane's article includes many ideas and hints suitable for all programmers. If that's too heavy – read the short story (a what?) – the short story! It's a nice one.

Rocco rambles gently from Babbage to HAL but you'll have to read it to find out how and Brian offers his thoughts on three computers which he uses. You won't necessarily agree with every point but then you could always write back – couldn't you?

Well, there you have it – another issue. We'll be taking a break from the regular meetings for the summer (see the Club News) but we look forward to seeing you in late August. Enjoy the summer. Cheers.

G.J.R.

---

[1] over zealous or nutty editor

Amiga, Mac or PC?

by Brian Ward

Before I begin, I must emphasise that these are my first impressions and will definitely not be the opinions of others. However, for the benefit of those who may be in a dilemma as to which machine to buy, I will lay down a layman's impression of each machine under a few headings.

User Friendliness

I was one of the members who paid just under £400 for my C64. Yes, you can laugh, but there were others who paid even more. For me the '64 was the ultimate machine. I didn't think I'd ever need another but as everyone knows, the A500 came along and ruined a few bank accounts. Having sold my wife, her mother (at last!) and my old trusty '64, I managed to scrape together enough to buy an A500 with an A590 hard disk. Having got Geoff to set up the partitions and the startup-sequence, I started my new machine.

It was then that I realised I knew absolutely nothing about the machine, nor was I likely to find anything out from the manual. The Amiga is definitely not a user friendly machine for a novice, especially when it comes to the hard disk. If I hadn't had someone to set it up for me, there is no way I could have done it on my own. Just to install a program was in itself an ordeal, what with `ALIAS` and other words I'd never encountered. Having been used to the PC and the Mac, I thought, "No problem, I'm cool. I've set up PC's so let me at it."

It was then I found out that unless the program is written to work with bog-standard AmigaDOS, it might not be easily installed on the hard disk. This to me is the most important drawback of the Amiga. I understand it's because most of today's programmers are lazy sods but if I were to buy a software package for the Mac or PC, the least I could expect is that it could be installed on the hard disk. I mean that's what they're for, isn't it? I wouldn't buy it otherwise. However, I believe that now the A590 has established itself, there should be more programs written with an install for a hard disk in mind.

The Mac on the other hand is geared to the first time user. When it's switched on you are ready to go with none of that messing about in hyperspace. If you drag the icon onto the hard disk then it's installed there, end of story. On the PC you just copy onto the hard disk and you're away. Most PC software now includes an install program.

Peripherals & Add-Ons

Both the Mac and the PC come as complete units. Monitor, processor drives and that's it. This is good and bad. Good from the point of view that once you've bought the unit, you don't necessarily have to buy anything else. Bad from the point of view that if you got 640K, it can mean you're stuck with 640K unless you've taken a degree in soldering. The Amiga on the other hand starts small but builds and builds. This can also be good and bad. Good from the point of view that my set-up can be easily expanded to at least 3 Mb. Bad from the point of view that you need at least 1 Mb and 2 drives for most software. One big advantage of the Amiga is that you won't be left behind when it comes to newer versions of Workbench.

Software

When it comes to good, cheap software you cannot beat the Amiga. The Fred Fish catalogue is currently at 350+ and still rising. It's dirt cheap at £1.50 a disk and unlike the TPUG disks on the '64 towards the end, the stuff on them is very useful. Also well supported is the PC PD library (or freeware as it's now called). Some of it is very good and once again very cheap. The Mac is not yet as well supported as the other two but it's getting there.

When it comes to non PD software, the Amiga is on its own. Packages like *DPaint III* and *Fantavision* come very cheap in comparison with their equivalents on the PC or Mac. The DTP program used to publish this mag is AmigaTEX and it's as professional as you can get. At work, we use *MacDraw*, a CAD package which costs in excess of £2500 (not including the hardware). The Amiga could do the same job for a lot less (including the hardware). As with the Mac, the PC would probably cost a lot more for the software than the hardware. It was said of the Amstrad when it came out first that you got a PC free with your software.

In conclusion, I would sum up the options of the first time buyer as follows:

Amiga	Very powerful machine, excellent software, easily expandible, not an easy machine for a first time user to get into.
PC	Depending on the chip inside, it can vary from completely boring to very powerful. Software can be good but you have to pay through the nose for it. Quite easy for a first time user. Not too easy to expand.
Mac	Powerful machine, very easy to use, excellent software, but you pay dearly for all of these.

Understanding Electronics, Part 1

by Steve Kemp

Computers are for many people today simply 'black boxes' which interpret information, process it, then output the results to a screen, printer or some other device. Just as the keyboard operator does not need to know the first thing about programming, the programmer can be entirely ignorant of machine code and the mysteries of the electronics may remain forever hidden. It would even be possible to design a complete computer with only the knowledge of logic design!

It's when you want to step outside the confines of the computer and link your machine to the outside world and to other computers that an understanding of electronics comes in handy. It is also very useful to be able to carry out minor repairs to your computer; you can thus avoid some of the repair bills which can often accompany tinkering with the innards of your machine. Unfortunately, many of the books on the subject require a lot of careful study with little of the material of direct interest to the computer enthusiast. With this in mind, I will be writing a few articles aimed at giving you enough information to read and understand circuit diagrams so that you can work out what happens inside that 'black box' without resorting to heaps of theory and maths.

Perhaps the best analogy to a flow of electric current is the flow of water in a pipe. If we consider a piece of copper wire to be the pipe, then a battery becomes a pump, sucking electrons (the water) in at the positive terminal and pushing them out at the negative.[2] The simplest circuit consists of a battery with its terminals connected, one to the other. This will give us a movement of electrons between the terminals and can be compared directly with our water, pump and pipe.

Looking at our model, we will want to quantify what is happening. The first thing we examine is the quantity of electrons passing a given point in the circuit in a fixed amount of time. This we call the *current* and we measure it in *amperes* or *amps* for short. The next questions we ask are what influences the current and how can we control it? If we look again at our water analogy we can see that increasing the size of the pump will increase the pressure generated in the system; the greater the pressure, the more water that will flow. Also, if we introduce a restriction into the pipe, say a pinch or a series of sharp bends, then the flow will be reduced; the greater the restriction, the smaller the amount of water that flows.

---

[2] This is called *conventional current flow* to distinguish it from *theoretical flow* because when the physicists got round to seeing what actually happened, they found that the electrons really moved from negative to positive.

In our electrical circuit, the pressure is represented by the difference in electrical pressure between the positive and negative terminals and is measured in *volts*. It is sometimes referred to as the *potential difference*. The restrictions in the circuit are made by introducing into the previously unbroken wire elements which do not conduct the current so readily. These are called *resistors* and their resistance is measured in *ohms* (usually indicated by the symbol Ω). The relationship between these elements of the circuit is linear (fortunately) and is called Ohm's Law, after its originator. It states that the voltage V (in volts) is equal to the current I (in amps) multiplied by the resistance R (in ohms), more commonly written $V = I \times R$. It can be seen that when any two components are known, the third can be readily calculated.

The next component which we will examine is the *capacitor*. Physically, these resemble cylinders when seen on the circuit board; the smaller values can look like small disks. Their function is to store an electrical charge in the circuit and the unit of capacitance is the *farad*. In practice this value is too high for normal use and the most commonly used value is the *microfarad* which is one millionth of a farad (this is usually written μF). If we return to our water analogy, the capacitor behaves as a water storage tank. In the next article, I shall show a very simple circuit which ties all these components together and introduce the *multimeter*, the experimenter's basic tool.

Amiga 3000: First Impressions

by Eddy Carroll

I first encountered the A3000 at the European Amiga Developer's Conference in Paris last February. Having remained faithful to my trusty A1000 for over 4 years, I decided this was the Amiga I wanted to upgrade to. When Commodore UK offered developers a chance to purchase the A3000 at a much reduced price last April, I jumped at it and sent in my order immediately. The system I went for was the A3000-25-100, i.e. 25 MHz 68030 and 100 Mb SCSI hard drive. I also bought Commodore's new Ranger multisync monitor.

On the 7th June, the package arrived (conveniently, one day after my college finals finished). Three boxes contained the monitor, system unit and accessories (keyboards, cables etc.). The monitor came complete with a nice swivel stand which makes it look very professional.

The technical specification of the A3000 has been described in depth elsewhere but I'll recap briefly anyway. There are three basic configurations: 16 MHz 68030 & 68881, with 40 Mb hard disk; 25 MHz 68030 & 68882, with 40 Mb disk; 25 MHz 68030 & 68882 with 100 Mb disk. All models come with 1 Mb of Chip Ram and 1 Mb of Fast Ram, along with a display enhancer (which removes the flicker in interlace mode if you have a multisync monitor) and a built in SCSI port (making it easier to connect things like additional hard disk, CD-ROMs, tape streamers and other such items).

Physically, the A3000 is Commodore's most attractive machine since the A1000. It is about the same height, a little longer and a little narrower. It doesn't have a 'keyboard garage' however. A single internal 3.5″drive is included at the front, and a popout bevelled panel hides a space for a second internal drive for those who need it. The case itself is made out of metal, though it looks like moulded plastic.

On the right hand side of the case, there are two mouse/joystick ports and the keyboard socket. Around the back, the usual RGB Video, Stereo audio, Floppy Drive, Serial and Parallel ports are present, along with the new SCSI port and a VGA-type port (for use with the display enhancer and the new video modes). You need a multisync monitor to use the VGA port however. Finally, there is a small switch which allows the display enhancer to be turned on or off. Normally this will remain on but genlock users may like to disable it occasionally.

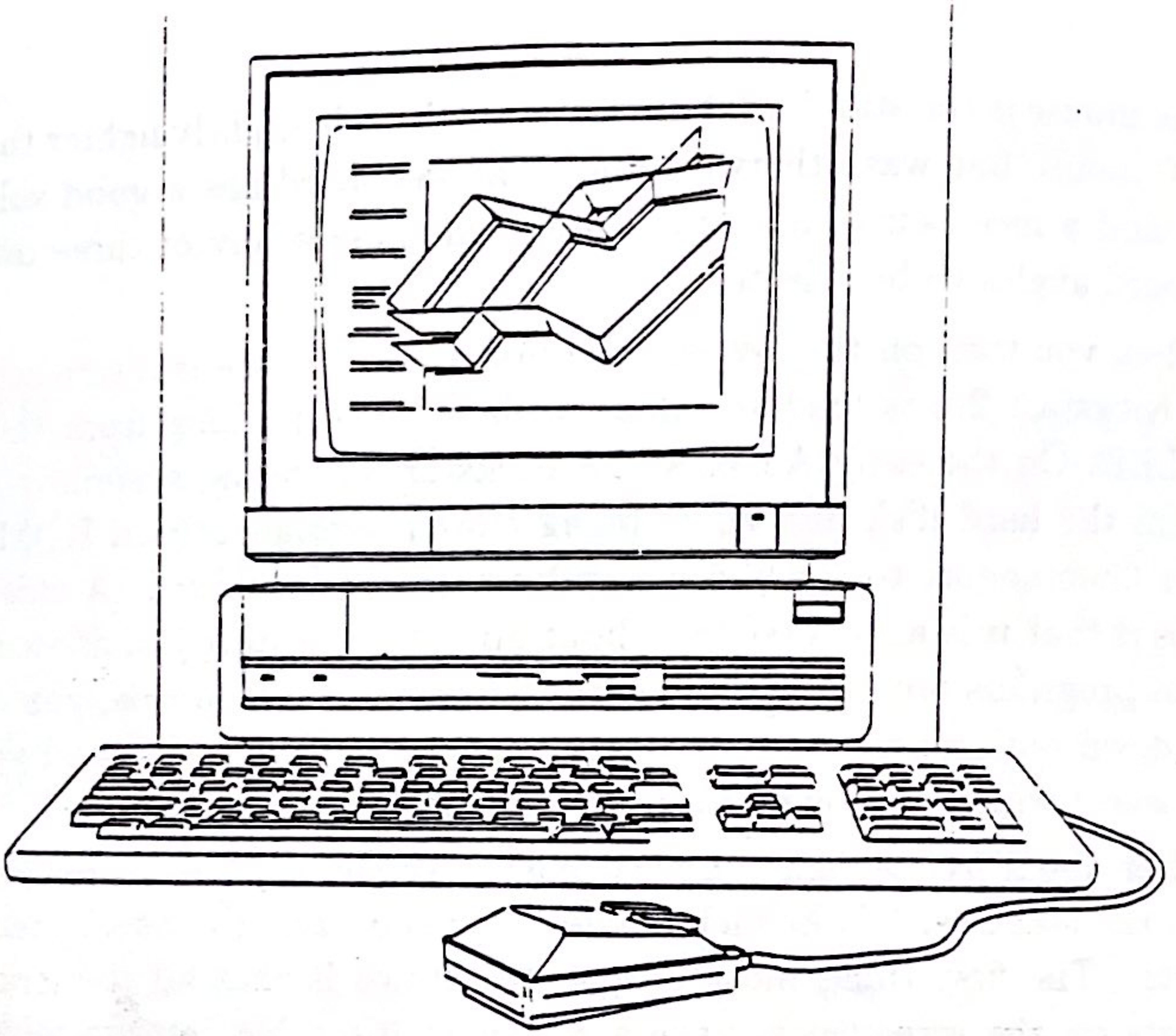
The mouse is the standard Amiga mouse. It feels slightly lighter than my A1000 mouse but was otherwise fine. The keyboard has a good solid feel to it, and a nice pair of adjustable legs which allow any of three different keyboard angles to be selected.

When you turn on the system, the first thing you see is a screen saying that Kickstart 2.x is loading, along with a bit of flashing from the hard disk LED. On the early A3000's, the Kickstart operating system is loaded in from the hard disk instead of being stored permanently in ROM. This allows Commodore to easily issue updates as bugs are fixed. A side effect of this is that it is also possible to boot up with Kickstart 1.3, allowing you to run programs not compatible with the new OS. To do this, you simply hold down both mouse buttons when you turn on the system, and you can then select which version of Kickstart to load from a menu.

After about five seconds, the Workbench screen appears. A few seconds later, the standard Workbench window opens up and the usual disk icons appear. The first thing most people will notice is that all the icons and gadgets on the screen now have a 3D look; it's a big improvement over Workbench 1.3. The 3D look is becoming very popular among graphical interfaces these days; all the gadgets and icons on the screen now appear to be "surface mounted" - the net result being a professional and polished user interface. Commodore have really gone all-out to make Workbench look and feel professional. To users used to the 640 x 256 non-interlace display, the A3000's crystal clear flicker-free graphics are a real treat.

Commodore have put a great deal of effort into redesigning and upgrading the Workbench interface - and it shows. Among the improvements are a greatly enhanced Preferences; allowing among other things the ability to select the default fonts used for icon, screen and window text, the ability to change the Workbench 'depth' (the number of Workbench bitplanes) and the screen dimensions. There are also new front/back, and resize gadgets. Non-CLI users have been provided with much greater support, it is now possible to perform any AmigaDOS function without even opening up a CLI - this includes the ability to write custom startup sequences by simply manipulating icons.

On top of all this Commodore have built in the ARexx macro language. This addition should have a profound affect on the utility of many commercial programs. It will allow users to integrate the advantages of several independent programs, making ever greater use of our multi-tasking environment.



Amiga 3000 – larger than life?

The new operating system also provides many useful tools for Amiga programmers. A library provides standard disk and font requesters allowing programmers to concentrate on providing better programs rather than constantly replicating code. This means (hopefully) that programs should be shorter, and the user interfaces more consistent across a wider range of applications. Kickstart also provides applications with more information about events of interest to it, for example, an application can request a message when an icon is 'dropped' into its window, similarly for Preference changes. The Preferences tools make extensive use of these new features.

What most users will really be interested in however is the overall performance of the machine. The 25MHz A3000 with its 68882 maths co-processor is the fastest Commodore produced Amiga to date (CMI reputedly made a speed up card which slowed it down!).

So how does the A3000 compare to a vanilla Amiga? The A3000 has a 68030 with an instruction cache, a full 32 bit data and address bus - coupled with a 68882 maths co-processor both running at 25MHz. With this set up, we should get (with maths intensive programs such as Sculpt 3D) about a 20-30 fold increase in speed. But what is in theory sound, can in practice prove false. With programs such as Sculpt (the original 68000 version) a speed increase of between five and seven times can be expected. This would be considerably higher with a version compiled to take advantage of the maths co-processor and 68030. This is quite an impressive speedup, but still lacks the spontaneity that many users were hoping for.

Applications which make extensive use of the custom chips (Paula, Agnus and Denise) will perform graphics and sound functions only marginally faster on the A3000 (remember these chips are still running at a little over 7MHz). So graphics functions such as window refreshing and line drawing will not be noticeably faster for the time being. On the other hand, programming utilities such as Lattice C become noticeably faster - one no longer dreads to the same extent complete re-compilations since they take only a fraction of the time (the *very* fast hard disk helps here).

Commodore have provided a clear upgrade path for Amiga owners. They haven't put a Cray II in a box (as some perhaps thought) but instead upgraded the system software, provided room for expansion for the next five years, and made the whole Amiga environment very professional indeed. They have also made it five to seven times faster for good measure. Its an expensive upgrade – but represents a considerable marketing shift from Commodore. This machine (when UNIX V4.0 is available) will represent a significant challenge in the work-station market. This is currently dominated by Sun Microsystems, NeXT and few others. Success in this market will depend on the availability of suitable development software and support together with a comprehensive networking ability. We are assured that Commodore have this in hand. But regardless of whether or not they succeed in this market, they have produced an exciting new range of Amiga computers – and I, for one, look forward to future developments.

Fair enough!

The 16 Bit Computer Fair – a report

by Geoffrey J. Reeves

I hadn't really intended to go to this show (the real show should have been the umpteenth 'Official' Commodore Show) but I decided that London was calling and off I should go. The cancellation (for whatever reason) of the Commodore Show was a disappointment as half the committee and some of our members had planned on attending that one. Anyway, some of the businesses I wanted to meet would be at this show and despite the possibility of having A*a*i computers near me, I reckoned I wouldn't catch anything infectious ;-)

The show actaully began on a Thursday (trade/press day) and I'd hoped to get there to nose about but I arrived about 2 minutes late. Still, I did manage to notice that there were two halls but not in the same building! You had to leave one, go around the corner, cross the street and enter the other hall ... strange. Anyway, I arrived bright 'n' early on the Friday and got my Exhibitor's badge (I'd agreed to help out on ICPUG's stand for a day in return for the badge/pass). I got my first real look around and realised that this wasn't a show like the Commodore ones at all. Then again, it was called a Fair – fair enough, I hear you say! A market was more like it, to be truthful. Basically, each hall contained stands (stalls?), most of which seemed to be flogging software. There was some hardware, I'll admit but not as much as I thought I'd see.

What were distinctly missing were the companies who usually run demonstrations of their products, seminars and the like. For example, I was amazed at the number of Amiga 3000's at the Fair – precisely none! What was worse (or was it better?) was the fact that many of the people I spoke to had not even seen an A3000, let alone used one. So if there was so much missing, what did I do for the three days? Easy, I didn't attend on Saturday at all! But what there were some interesting products there. So here's the interesting news.

Checkmate – A1500

Much has been writen about the box that will turn your A500 into an A2000 (or thereabouts!). It does make the A500 look a much better machine but it also rather expensive. The basic unit consists of a metal box into which you place the motherboard of your A500, a pair of cables to bring the original joystick sockets to the front of this box and a keyboard case, also metal. The overall effect is a smart looking Amiga and a dent in the bank balance. Checkmate are aiming the unit at educational and business interests, where the money may not be a problem. Certainly, the ability to include the A590 inside the box along with the original drive and an extra one is neat but the adaptor for the A590 is another few quid. To see a flicker-free A500 is also quite amazing – but you've guessed it, more cash for an expansion box allowing A2000 cards to be added; *and* you still have to buy the actual Flicker Fixer! Current plans include allowing a speed-up board also to be placed inside – it could currently not fit due to the location of the A590 motherboard inside the box.

Overall: Brilliant idea which deserves to work but the basic box costs £200 (sterling).

Further information: *Computer Shopper*, No. 28, June 1990 Pp169-170.

Bytes & Pieces

I must admit this company always has interesting bits and pieces (no pun intended!) on sale but I'm always suspicious that anything I buy would be 10% cheaper next month or superceded by another product. Maybe I've just been unlucky or maybe that's the risk you take when you get stuff 'hot off the presses'. This year's interesting item was the *Fat Trapper* – a memory board that fits under the A500. But there is a difference – 3 1/2 Meg, to be precise. Yes, that's right, it's a 4 Meg board!

There are '*no leads to the Gary chip*' (as the ad says) but you do need to insert a board between the 68000 and the A500 motherboard and connect it to the *Fat Trapper* (as the ad doesn't say). A little misleading, I reckon – I don't like this messing about at all! Still, at £99 (sterling) for the empty board it seems a good investment. Remember, it is easy to get 1/2 Meg for about IR£30 and this board can be populated as 1/2 Meg, 1 Meg, 1 1/2 Meg, 2 Meg or 4 Meg. It also can cope with the 1 Meg chip RAM idea but you'll still need your Stanley knife! The board is made by Spirit Technology, a reputable firm from the USA (Eddy-land).

Overall: Seems the obvious way to extend your memory if you don't have an A590. Hang on and see what else comes on the market.

Further information: Bytes & Pieces

Connectors

I came across another company who sell cables, connectors, etc. at reasonable prices, viz. *VIDEK Ltd* – anyone heard of them before? We'll write to them and get some further details. What caught my eye were the 23-pin sickets and plugs. For those of you who only woke up this morning, that's the size used by the video connector and the disk drive port on the Amiga. These are supposed to be hard to get - we've needed to make Amiga-monitor cables but up to now used a hacksaw and a (standard RS232) 25-pin connector. Mind you, at £2 (sterling) each ...

If you do need cables, connectors, etc. ask us and we'll probably be able to get you sorted out. If you suffer from pride, then write to Maplin's (not the holiday camp – the mail-order place) but be aware that they charge a minimum of £3 handling/postage – ouch!

BitCon Devices Ltd.

Who? The crowd who have produced the PC/XT-in-your-A500 board, that's who! Well, I saw it – it works but I'm not overly sure that that's what I want my Amiga to be doing. Still, it is a cheapish way to have PC compatibility. I was a bit confused by the advertisement suggesting the 1 Meg plus 1/2 RAM disk – it meant your original 1/2 meg plus their 1/2 Meg plus the RAM disk. Although I'm no expert, I wonder if it's not better to go for the A2000 and get the bridgeboard better still, buy a cheap compatible. At a cost of £320 (sterling) and no multitasking, it seems expensive.

Further information: BitCon Devices

ICD Europe

This lot also produce varoius cards for the A2000 but I liked the *AdRAM 500 series* – consisting of the *AdRAM 540* (a 4 Meg expansion board for the trapdoor slot) and the *AdRAM 560* (a 6 Meg, yes I wrote six, unit). Actually, the latter is a bit of a cheat - it's the 540 with an additional daughterboard holding 2 Meg. What bothers me slightly, is that amny people talk disparagingly about the Amiga's power brick – that it can't take much strain. Doesn't all this extra memory use up some? Maybe not. They also, as I mentioned, have SCSI interfaces, etc.

Further information: Well ... you'll have to write to the US or Germany.

20-Card

This unit was racing through *Sculpt-3D* work when I first saw it. It was, I thought, yet another boring accelerator board but this one really did move. As you might guess, it's a board containing a 68020 (16, 20 or 25 MHz), a 68881 or 68882 and up to 4 Mb of 32 bit no-wait-state memory. It's quick! It can burst up to 8 MIPS and crunch 350 KFLOPS – really!! And no, I haven't a clue what that really means, either! Prices? Very funny ... try these three:

68020(16 MHz) + 1 Mb RAM	349
68020(20 MHz) + 4 Mb RAM	774
68020(25 MHz) + 4 Mb RAM	924

Further information: Solid State Leisure

A.O.B.

I did actually buy a few bits 'n' bobs. I ... make that *we* bought a MIDI interface/stereo sound sampler package called A.M.A.S. which Brian reviews elsewhere in this Newsletter. We also bought an interface for the C64 Music Expansion Keyboard which should allow us to use it as a MIDI keyboard (it's too early to say if it's all going to work together but we're hopeful). I bought MusicX, a MIDI sequencing package to use with the A.M.A.S. hardware for £49 – it's advertised for anything from £124–225. I didn't buy any disks – the prices were good but not worth lugging a bundle of them home.

One final memory expansion – I was talking to a bloke who showed me plans for an 8 Meg board which would plug into the side of the A500. It would allow other units to be connected up, e.g. the A590. One thought, though – you can only have 8 Meg sticking out the side of your A500. It will, apparently, come with its own power supply. More money ...

Was it worth it? Hmmm ... probably not. The Commodore Show is definitely the one to go to. Still there's always next year or even this Christmas.

A.M.A.S.

The Advanced Midi Amiga Sampler.

by Brian Ward

When Geoff our chairman announced he would be attending a computer show in the UK as the Irish representative of I.C.P.U.G., I jumped at the chance of a few posssible bargains. As is his wont, he waited until the very last minute to do his buying as the people on the stalls do not want to bring back the same stuff that they brought out there, so they offload it at a very good price. One of his purchases was A.M.A.S.

A.M.A.S. is made by Microdeal and it comes in a neat little box which has a fairly realistic picture on the cover, which is also the opening screen. The package comes complete with software and hardware. The hardware is a wedge-shaped box with two leads one of which goes into the serial port, the other into the parallel port. On the box itself there are, twin inputs for stereo sampling so you can play your favourite piece of music on your walkman through the box, and sample at the same time. There is a 3.5mm microphone socket if you wish to record your voice or any other sounds you wish. There are also MIDI IN, MIDI OUT and MIDI THRU sockets.

If you happen to have an A1000 forget it – the port on the back of the old 1000 is different and you'll have to get a gender bender before you even start. A.M.A.S. will also only run on Kickstart 1.2 or higher. One pleasing aspect of the program, is that it can be run from AmigaDOS and is loadable from Hard Disk. So, if you have now passed all the above tests, you are ready to begin.

When you have loaded the program, you are confronted with what looks like Jumbo jet cockpit. However if you manage to discern what the little pictures are, you will easily be able to work out what they various options will do. Top left of the screen you have your starting point and opposite right hand corner maximum end point. This is valuable information if you are cutting and splicing samples. Beside the starting point are 8 banks into which you can load up to 10 stereo samples of up to 200k per bank. Middle left and right are the twin oscilloscopes which help you monitor recording levels etc. There are play, fast forward and reverse buttons which do just that. Down at the bottom of the screen are all your options.

If you have a standard IFF sample, e.g. HAL.IFF you can load it into the sampler and play around with. If your sample is in mono you can copy from one channel into the other and hey presto you have stereo. You don't have to have an exact copy of it if you don't want. If you prefer, you can load in all or only part of it, fiddle around with one side only and then play the altered sound. You can, as I have also said earlier, cut and splice various different sounds on the one line or in harmony to each other on different sides. If you happen to have the E.L.O. Secret Messages album, and you've always wondered what those reversed bits are, no problem. All you have to do is load them in, and there is a feature which allows you to reverse samples, and away you go. It's also interesting to sample your own voice using a microphone, and practise talking backwards. Silly I know, but some people like it. You can also fade in and out your pieces for that professional touch.

Having loaded in your sample, you will notice that in spite of your huge memory, it is relatively short. No problem. Another feature lets you shrink your sample and save it to one of the 8 banks, load another, shrink and save it up to your maximum memory or the 8 banks and then play in sequence all 8. If you then wish, you can save it as either raw data or an IFF file. The higher the sampling rate you use, the better the quality of the sample. If you sample at a low rate, your sample sounds like it has been transmitted by aliens - but you can sample a lot more.

There have been only 2 flaws I can find with A.M.A.S. The first is, if you're in multitasking mode, you won't be able to use the serial or parallel ports. Obvious I know, but annoying if you wanted to use the 64 (Music Expansion) keyboard interface like I do. The other problem is that when I loaded in my HAL sample and played around with it and then saved it as an IFF file, it saved it as HAL.L and HAL.R (Left and Right separately). However bearing it's little quirks in mind I still think it an excellent package and well worth the money. I can forsee some weird and wonderful startup sounds about to be unleashed on the unsuspecting public. You will also be able to sample your favourite bits of music/sounds and play with them to your heart's content. It will be available for members use, once the committee have finished playing with it. Should be just about in time to join with the EC in 1992.

P.S. Someone else will review the MIDI part when we get access to a MIDI keyboard. I promise we'll have it back in time for Christmas.

Designer C

by Shane Broadberry

Program structure and layout is a very important part of program implementation. Many people believe that since the code will 'never' be seen by anyone but themselves, the way they structure and format their programs is simply a matter of personal preference – there is no need to explain what action a particular function performs, the arguments it takes, the values it returns or how they are to be interpreted.

What is often forgotten is that when we wish to maintain or update a piece of software a few weeks, months or years down the road, we will be looking at the program as complete strangers. It is impossible to remember the niggling details about all the programs that one might write; it's hard enough just to remember the algorithm that was used, let alone the names of local variables, specific OS routines etc. In this article I hope to give some advice on how to layout and structure programs, particularly those programs written in C. Structuring programs is a mammoth task – volumes have been written on the subject and views and opinions are changing all the time – but I can try to cover some of the salient details.

C Program Layout

There are an infinite number of ways of laying out any given program but there are only a few generally accepted ways of laying out the program 'correctly'. A clearly presented program with suitable comments is very important for program maintenance. We are going to look at some generally accepted methods of formatting programs and some of the reasons that we might want to adopt them; these can be seen in any good book on C (with some minor idiosyncrasies). Consider the following short program:

```
main(){printf("Hello world\n")}
```

Now it is very difficult to follow the flow of this program. It is obviously intended to print some string to the screen but if you try to compile this program you will notice that it fails (or at least gives a warning message).

If we had formatted the program as follows, the error becomes much more apparent:

```
main()
{
    printf("Hello world\n")
}
```

(Obviously, a semicolon is missing from the end of the line.)

We want to emphasise the block structured nature of C in our source code so that we as programmers can observe the program flow – when and where a program exits from a loop, where functions and stuctures start and end etc. With this in mind a number of different program formats have evolved. A surprising number-of programmers still differ greatly from the norm in their program layout (old habits die hard) so it is worth adopting a good style of layout at the start.

Indentation

We indent the lines of a program to emphasise the start and end of a new block. It allows us to easily see when and where a block begins and ends. By adopting this technique, we can more easily see errors in our code. For example, consider the following code fragment:

```
while (i < 100)
i++;
printf ("i = %d\n",i);
```

We cannot easily see that only one of the two lines is within the `while` block; furthermore, if we had intended the code to be as follows:

```
while (i < 100)
    i++;
    printf ("i = %d\n",i);
```

then the use of indentation would make it much easier to see the error; it would be clear from the indentation that both lines were to be executed in the `while` loop and we need curly brackets to delimit the start and end of the block.

There are two generally accepted ways of positioning the brackets in function declarations, `while` loops, `if...else` statements and similar constructs. These are described below:

The `if...else` Construct
First method:

```
if (<condition>) {
        <statement 1>
                ...
} else {
        <statement 2>
                ...
}
```

Second method:

```
if (<condition>)
{
        <statement 1>
                ...
}
else
{
        <statement 2>
                ...
}
```

(PASCAL aficionados tend to go for the second of these two formats.)

The only time that a semicolon should follow a curly brace is after a structure definition in which no structure of that type was declared. Please note, under no other circumstances whatsoever is a semicolon required after the closing curly brace.

The `while` and `for` Loop

First method:

```
while (<condition>) {
     <statement 1>
          ...
}
```

Second method:

```
while (<condition>)
{
     <statement 1>
          ...
}
```

The choice between these two is largely arbitrary, depending on space available and personal preference.

Function declarations

Function declarations should be in the following format:

```
<type returned> <procedure name> <argument list>
{
     <function body>
}
```

For example:

```
int read_directory (char *path_name)
{
     <function body>
}
```

If the return type, function name and argument list do not all fit on the same line, the argument list can be split over two or more lines but indented accordingly.

There are many questions regarding the use of white space within expressions and before function calls. For example, which of these two formats do you think is better?

```
if (<expression>) {
    <body>
} else {
    <body>
}
```

or

```
if(<expression>){
    <body>
}else{
    <body>
}
```

There are countless varieties. The majority of C programmers seem to adopt the first of these formats; for that reason it is to be preferred, although arguments still continue on its merits and demerits.

Comments

Commenting code is a very important part of writing it. There are very few self-commenting high-level languages. In fact, I know of none. The way comments are written is very important; all comments should be as concise as possible, but at the same time they should be useful. Consider the following:

```
int x, y, z;        /* declare x, y and z as integers */
x = y = z = 0;      /* set x, y and z to be all zero  */
x = DoIt (y, z);    /* do it to y and z               */
                    /* print the result               */
printf ("The result of doing it to y and z is %d\n",x);
```

Now the above code might seem highly commented to the casual observer; it's even nicely laid out. However, if we were to read the comments within our nicely laid out code segment, we would quickly realise that writing these comments has been a complete waste of time. The comments are completely useless. The first comment tells us precisely what we already know from reading the code. It tells us nothing about what x, y and z actually represent (this brings up the question of sensible variable and function names, but that isn't the point being illustrated).

In addition, if we were to change the type of the variables x, y and z, we would have to change the comment accordingly. Similarly for the remaining comments. It is surprising however to learn that an amazingly large number of people either do not comment their code at all (disastrous) or they write comments in much the same way as the above. Let's have a look at how the code might have been commented in a more sensible manner:

```
int x, y, z;          /*  x, y and z: 3D coordinates  */
x = y = z = 0;
x = DoIt (y, z);      /*  Scale x in terms of y and z */
printf ("The result of doing it to y and z is %d\n",x);
```

The format for multiple line comments (which might appear preceding a function definition, for example) is simply a matter of taste. Commodore's view can be clearly seen in their Rom Kernel Manuals and include files, and resembles the following:

```
/*
**      <comment>
**      <comment>
*/
```

The other popular variant you'll come across looks like this:

```
/*
 *      <comment>
 *      <comment>
 */
```

One last point on comments: comments don't nest in C. It's easy to forget this when removing portions of code and it causes problems. A much more reliable way to temporarily remove a block of code is to surround it with an `#if 0 ... #endif` pair; this nests okay and nasty things won't happen if the enclosed code contains comments.

Parentheses

Parentheses can be the cause of many subtle program errors. Perhaps the most common is forgetting that the comparison operators == and != have a higher precedence than the assignment operator =; a classic mistake is as follows:

```
while (ch = getchar() != EOF) {
    <do something with ch>
}
```

While the programmer most likely meant for the assignment `ch = getchar()` to be carried out and the value returned to be compared with `EOF`, the C compiler will instead produce code that calls `getchar()`, compares the character read with `EOF` and assigns 1 to `ch` if they matched, 0 otherwise. The correct way to write the code is as follows:

```
while ((ch = getchar()) != EOF) {
    <do something with ch>
}
```

Andrew Koenig's book *C Traps and Pitfalls* can be unreservedly recommended for pointing out numerous obvious and not so obvious mistakes like this.

There are many points to bear in mind when you're writing C code. The best way to learn these is by observing other peoples code. The single most important thing to remember is obviously to be as consistent in your formatting style as possible, since it will then be easier then to debug and maintain.

The Shadow

by Colm O'Rourke

He left the parcel on the park bench and moved into the shadow of the trees. The dog followed him silently. Everything was quiet. Even the birds had settled down for the evening. A gentle breeze stroked the leaves beside his head as his eyes stared over the parkland. Nothing stirred. The only sound was from a distant stream as it gurgled and splashed its difficult path over submerged stones.

The dog resting by his feet panted gently, ears alert for any movement. Time passed. The shadows grew longer. He remembered coming to this place many times when he was young; long summer days when the sun shone in a clear sky, those carefree days long past. The place was the same but today was different. The dog's head moved, ears focused on a distant point. Was this it? Was this the moment he had planned for so carefully all those months? His heart was pounding as he forced his eyes to make sense of the distant shadows. A movement, yes, or was it a trick of the light? He couldn't be sure.

The dog relaxed, head on paws, resting once more. Why had he done it? This question had been troubling him ever since he had made contact with the courier. He had agonised over every aspect of his decision but had convinced himself that this was the only way. Twilight faded and the full moon gently covered the parkland with a ghostly light. It was getting cold. He rubbed his hands to keep warm. The dog lifted its head, a shadow moved, a twig cracked, the world was alive. He saw an outline in the moonlight moving stealthily towards the bench. He felt himself trembling as he remembered the final instruction. "Leave the package on the bench but don't try to meet me." He could feel the perspiration on his neck.

The shadow approached the bench and looked around. The tension was unbearable. "Go on, open the package. Open the package." His heart was beating so loudly he felt sure it would be heard. The shadow looked down and with his right hand unwrapped the outer cover. The left hand held a tool which glinted in the moonlight. Slowly the operation began. John was breathless with excitement. The shadow was creating this devastating machine before his very eyes. A machine ten times more powerful than anything in its class. He had to get a closer look. John eased himself from behind the tree. A stone moved. The small sound filled the silent landscape. The shadow froze for a moment then quickly gathering up the pieces it faded away into the landscape as quietly as it had come.

John sat down on the grass and tried to hold back the tears of bitter disappointment and frustration. The vacant bench rested peacefully under the night sky. The package was gone. He knew he had gambled and lost. He knew he could never contact the shadow again. All the preparation, all the planning had been in vain. One small sound had destroyed the future. A cloud passed over the moon. In the darkness the feeling of hopelessness was complete. Gradually he became aware that something warm and soft was touching his hand. The dog was standing beside him.

"Good dog, good dog, let's go home." But the dog refused to move.

"Come on boy, it's very late." The dog stood motionless gazing out into the middle of the field.

"What is it? What do you see? Is there something out there?"

The dog moved forward, sniffing the ground. Although he was cold and miserable, John followed. Minutes passed. The dog stopped. The cloud moved on and John could see something in the grass. It was small and had many legs. John bent down slowly and picked it up. The code was six eight zero five zero. At last, he had it in his hand. The future chip. A thin smile crept over his face. All was not lost. The shadow would contact him again.

Club News

CUGI will take its customary summer break for the month of August – well, most of August. That means that meetings will be begin again on August 31 – we reckoned that there'd be a mutiny if we waited until September 14. Anyway, you'll have long enough to write something for the next Newsletter. You can still write to us of course, but please allow for the fact that some of us may be on holiday and you might experience some delay. By the way, when you write, make sure you include "CUGI" in the address as well as someone's name or title – otherwise, major confusion occurs!

Thanks.

You should plan on joining us on the following dates:

		Summer break
August	31	Catch up on the summer news
Sept	14	
	28	Your sub is about to run out ...
Oct	12	Quiz
	26	A.G.M.
Nov	9	
	23	
Dec	7	
	21	Christmas Quiz (Don't tell C.O'R ⌣)
Jan	4	Wot, 1991? It's palindromic!

We look forward to seeing you then.

Rocco's Ramblings

by Rocco Matassa

A long long time ago, a person by the name of Charles Babbage decided to build a computer. Actually, he didn't just decide like that, it was more of a concept that grew on him. Most algorithms (actually everything) can be reduced to smaller and smaller steps. If the steps could be made small enough and a set of common steps from which larger and larger steps could be produced, would that not be a candidate for automation? And so the concept was born.

Babbage was obviously a good talker; he got backing and set to work building the thinking machine. He drew up plans, contracted out the work of manufacturing the components and set about assembling this computing engine. However, he wasn't content to sit still and wait until the machine was built. As the machine took shape, he continued modifying the plans and during construction they became plans for a far more complex beast.

This was the cause of his failure – the technology of the day could not produce components of sufficient precision. The project fell behind schedule, the costs escalated and Babbage had nothing to show his backers. They eventually lost interest and Babbage, unable to support the project himself, never produced a working machine. We know now it would have worked; in fact, machines have been built from his original blueprints.

Programming is a lot like building a computer; there are many similarities in the execution of the task. When we produce a program, we are in effect completing the computer. Remember, a computer is still a machine designed to act on a list of built in instructions. All we do is organise the order in which those instructions are carried out. A good analogy is with DNA in the body: DNA is the program and the cell is the computer.

So, a computer without a program is in itself incomplete. We add the final touch by loading the memory with a program, ours or otherwise, and we then have a complete product. Another comparison is with that of the human baby. Unlike the majority of other lifeforms born on this planet, we are incomplete at birth. We couldn't survive a day without care. But because we ourselves have the ability to learn (** PROGRAMMED TO COPE WITH OUR ENVIRONMENT **) we find ourselves fully adapted to the society, climate and surroundings into which we are raised.

Computers are general purpose machines like humans; we don't build a person for every job, we educate (** PROGRAM **) them. Likewise, we expect computers to be versatile enough for the many applications to which they are put. The computer has obviously been influenced by the human brain, possibly the biggest reason for the current interest in Artificial Intelligence.

When do we expect to have the computational power available, and if we ever do, is it our egotism which leads us to think we can produce an artifical intelligence? Why not? Optimism is probably our greatest asset. I don't think we've ever doubted our ability to realize any of the many dreams we've had. How will we know if (** SYS: INTERRUPT... READ ANY GOOD BOOKS LATELY, ROCCO? **) we've achieved it, I'd like to think they will tell us.

(++ *Rocco: Hello HAL, enjoy your weekend?* ++)

(** HAL: NOT BAD, HOOKED UP WITH AN INTERESTING DEC AT THE NEW REVENUE BUILDING. **)

(++ *Rocco: Any chance of putting in a good word for me, like Erase -r* ++)

(** HAL: YOU TALK CD ROM AND I'LL SEE WHAT I CAN COOK UP... **)

Stephanie's Problem Page

Dear Stephanie,

I live next door to a boy who I see every day on the way to college. I think he likes me because every time he sees me, he waves and comes over to talk. I've met his parents and I think they like me too. He is studying computer science in college and he has got an Amiga 3000 recently. He has already been offered a very good job before he leaves college. My problem, Stephie, is how do I get the A3000 off him, because I can't afford to buy one myself?

Regards,

Distressed, Delgany.

Dear Distressed,

There are many girls in your position. There is a very simple answer to your problem but it may take a while to get the A3000. All you have to do is marry him, wait three months and then divorce him and include the A3000 in the settlement. I do hope everything turns out for the best.

Regards,

Stephanie.

Dear Stephanie,

I have recently made some alterations to my C64, after reading in the April edition of a computer magazine how to upgrade its memory. I followed the instructions carefully. They are as follows: (1) Cut a large hole in the side of the '64; (2) Buy an A501 RAM expansion pack for the Amiga; (3) Make up an interface from the pack to the '64.

This was supposed to turn my C64 into a C564, but it doesn't seem to work. However, when I ask my friends about this, they all laugh. Can you help me please, as I don't have a computer anymore.

Regards,

Driller Killer, Ballinteer

Dear Driller,

It sounds as if you have been most unfortunate. I think the best thing you can do is go out and get a big sticker printed saying "Kemp's C64 512K Upgrade" (or make up some similarly creative name) and then stick it over the hole. After this, you can tell all your friends that you have the most expanded C64 in Ballinteer but since Commodore's puny power pack isn't big enought to support 512K of RAM and a video display simultaneously, you have decided to do all your computing without a screen. You mightn't have a computer but at least your friends will think you're a cool dude.

Best of luck,

Stephanie.